Mise en place de Seafile Community + OnlyOffice

Prérequis :

- Avoir une base de donnée `MariaDB` déployé en local
  - Voir déploiement MariaDB
- Avoir un nom de domaine
  - Dans notre cas le nom de domaine utilisé est : `miraceti.net`

Il existe plusieurs manière de mettre en place `Seafile` Docker. Dans notre cas nous allons utiliser une base de données `MySQL` déjà déployé avec Docker car nous allons utiliser la version community de Seafile.

Configuration nom de domaine

Grâce à notre nom de domaine, nous allons pouvoir créer des sous domaine (en CNAME `A`).

Sur le site de nom de domaine respectif, dans la section DNS, ajouter l'entrée suivante :

- drive.miraceti.net A ip.adress.server
  - Sera utilisé pour redirigé sur l'interface Web de Seafile
- office.miraceti.net A ip.adress.server
  - Sera utilisé pour éditer les documents avec onlyoffice

Configuration Docker Compose

> Pour accéder à notre base de données externe, nous devons importer le `network` lié à la base de donnée (dans notre cas le network est "`bddnetwork`").

Fichier `docker-compose.yaml` :

```
services:


  # Service de cache pour Seafile
  memcached:
```

```
  image: memcached:1.6
  container_name: seafile-memcached
  restart: unless-stopped
  networks:
    - seafile-net
  entrypoint: memcached -m 256

seafile:
  image: seafileltd/seafile-mc:latest
  container_name: seafile
  restart: unless-stopped
  depends_on:
    - memcached
  ports:
    - "8082:80"   # Port HTTP Seafile (interface web)
    - "8083:8080" # Port WebDAV (SeafDAV)
  environment:
    DB_HOST: mariadb
    DB_ROOT_PASSWD: myRootPasswordMariaDB
    SEAFILE_SERVER_LETSENCRYPT: "false"
    SEAFILE_SERVER_HOSTNAME: drive.miraceti.net
    TIME_ZONE: Europe/Paris
    MEMCACHED_HOST: seafile-memcached  # Nom du conteneur de cache
  volumes:
    - /my/path/to/seafile/data:/shared  # Emplacement des conf + data
  networks:
    - bddnetwork
    - seafile-net

onlyoffice:
  image: onlyoffice/documentserver:latest
  container_name: onlyoffice
  restart: unless-stopped
  depends_on:
    - seafile
  ports:
    - "8081:80"
  environment:
    - JWT_ENABLED=true
    - JWT_SECRET=MyJwtTokenSecretGenerateByOnlyOffice
```

```
      - JWT_HEADER=Authorization
    volumes:
      - /my/path/to/onlyoffice/data:/var/www/onlyoffice/Data
      - /my/path/to/onlyoffice/logs:/var/log/onlyoffice
    networks:
      - seafile-net

networks:
  seafile-net:  # Réseau interne de seafile
  bddnetwork:
    external: true
```

> Le mot de passe `DB_ROOT_PASSWD` est nécessaire seulement la première fois pour initialiser les bases de données et créer l'utilisateur.

Démarrage de seafile

Nous allons démarrer seulement le service `seafile` du docker compose pour initialiser la base de donnée et générer les fichiers de configuration.

```
docker compose up -d seafile
```

Les logs doivent ressembler à :

```
ocker compose logs seafile
seafile  | *** Running /etc/my_init.d/01_create_data_links.sh...
seafile  | *** Booting runit daemon...
seafile  | *** Runit started as PID 21
seafile  | *** Running /scripts/enterpoint.sh...
seafile  | 2025-10-09 16:48:48 Nginx ready
seafile  | 2025-10-09 16:48:48 This is an idle script (infinite loop) to keep container
running.
seafile  | [2025-10-09 16:48:48] Skip running setup-seafile-mysql.py because there is existing
seafile-data folder.
seafile  | [10/09/2025 16:48:48][upgrade]: The container was recreated, start fix the media
symlinks
seafile  | [10/09/2025 16:48:48][upgrade]: Done
seafile  |
seafile  | Starting seafile server, please wait ...
seafile  | Seafile server started
```

```
seafile  |
seafile  | Done.
seafile  |
seafile  | Starting seahub at port 8000 ...
seafile  |
seafile  | Seahub is started
seafile  |
seafile  | Done.
seafile  |
```

> Pour retrouver le mot de passe créé par l'initialisation des bases de données `Seafile`, il faut se rendre dans le fichier de configuration `seafile.conf` (voir ci-dessous)

> Si les logs ne commence pas de cette manière, cela signifie que l'image `seafile` n'a pas démarré correctement, peut être du à la connexion à la base de donnée.

Maintenant nous pouvons éteindre le service (`docker compose stop`) pour passer à l'édition des fichiers de configurations.

Edition des fichier de configuration Seafile

seafile.conf

Editer le fichier se trouvant au chemin `/my/path/to/seafile/data/seafile/conf/seafile.conf` pour y mettre la configuration suivante :

```
[fileserver]
port = 8082

[database]
type = mysql
host = mariadb
port = 3306
user = seafile
password = 123456789-b891-4071-8cf5-123456789
db_name = seafile_db
connection_charset = utf8

[notification]
```

```
enabled = false
host = 127.0.0.1
port = 8083
log_level = info
jwt_private_key = *****************
```

seahub_settings.py

Editer le fichier se trouvant au chemin `/my/path/to/seafile/data/seafile/conf/seahub_settings.py` pour y mettre la configuration suivante :

```
# -*- coding: utf-8 -*-
SECRET_KEY = "l*********************************@u#"
SERVICE_URL = "https://drive.miraceti.net"

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'seahub_db',
        'USER': 'seafile',
        'PASSWORD': '123456789-b891-4071-8cf5-123456789',
        'HOST': 'mariadb',
        'PORT': '3306',
        'OPTIONS': {'charset': 'utf8mb4'},
    }
}



CACHES = {
    'default': {
        'BACKEND': 'django_pylibmc.memcached.PyLibMCCache',
        'LOCATION': 'memcached:11211',
    },
    'locmem': {
        'BACKEND': 'django.core.cache.backends.locmem.LocMemCache',
    },
}
COMPRESS_CACHE_BACKEND = 'locmem'
TIME_ZONE = 'Europe/Paris'

# Security Settings
```

```
ALLOWED_HOSTS = ['.miraceti.net']

# Whether to use a secure cookie for the CSRF cookie
CSRF_COOKIE_SECURE = True
# The value of the SameSite flag on the CSRF cookie
CSRF_COOKIE_SAMESITE = 'Strict'
CSRF_TRUSTED_ORIGINS = ['https://drive.miraceti.net', 'https://office.miraceti.net']

# OnlyOffice integration
ENABLE_ONLYOFFICE = True
VERIFY_ONLYOFFICE_CERTIFICATE = False
ONLYOFFICE_APIJS_URL = 'https://office.miraceti.net/web-apps/apps/api/documents/api.js'
ONLYOFFICE_FILE_EXTENSION = ('doc', 'docx', 'ppt', 'pptx', 'xls', 'xlsx', 'odt', 'fodt',
'odp', 'fodp', 'ods', 'fods', 'csv', 'ppsx', 'pps')
ONLYOFFICE_EDIT_FILE_EXTENSION = ('doc', 'docx', 'xls', 'xlsx', 'ppt', 'pptx', 'odt', 'ods')
ONLYOFFICE_JWT_SECRET = '<replace_by_your_jwt_token>'
ONLYOFFICE_JWT_HEADER = 'Authorization'

# Mail configuration
EMAIL_USE_SSL = True
EMAIL_HOST = ''        # smpt server
EMAIL_HOST_USER = ''    # username and domain
EMAIL_HOST_PASSWORD = ''    # password
EMAIL_PORT = 526
DEFAULT_FROM_EMAIL = EMAIL_HOST_USER
SERVER_EMAIL = EMAIL_HOST_USER

# Webdav configuration
ENABLE_SEAFDAV = True
SEAFDAV_HOST = '0.0.0.0'
SEAFDAV_PORT = 8080
SEAFDAV_SSL = False
FILE_SERVER_ROOT = 'https://drive.miraceti.net/seafhttp'
```

seafdav.conf

Editer le fichier se trouvant au chemin `/my/path/to/seafile/data/seafile/conf/seafdav.conf` pour y mettre la configuration suivante :

```
[WEBDAV]
enabled = true
```

```
port = 8080
host = 0.0.0.0
workers = 5
timeout = 1200
share_name = /seafdav
```

> Démarrer le docker compose devrait rendre Nextcloud accessible en local, ou sur l'adresse IP de votre serveur sur le réseau local, sur le port `8082`.

Configuration Apache2

Création des nouveaux hosts dans `/etc/apache2/sites-available` avec les noms suivant `cloud.miraceti.net.conf` et `office.miraceti.net.conf`.

Nous allons nous concentrer sur la partie SSL de chaque host :

`cloud.miraceti.net-le-ssl.conf`

```
<VirtualHost *:443>
        ServerName drive.miraceti.net

        ErrorLog ${APACHE_LOG_DIR}/drive.miraceti.net.log
        CustomLog ${APACHE_LOG_DIR}/drive.miraceti.net.log combined

# Reverse Proxy Seafile (interface Web)
        ProxyPreserveHost On
        ProxyPass / http://localhost:8082/
        ProxyPassReverse / http://localhost:8082/

# Reverse Proxy WebDav
ProxyPass /seafdav http://localhost:8086/seafdav
ProxyPassReverse /seafdav http://localhost:8086/seafdav

# --- En-têtes nécessaires
RequestHeader set X-Forwarded-Proto "https"
RequestHeader set X-Forwarded-Host "drive.miraceti.net"

        # Reverse Proxy WebSocket (important pour notifications temps réel)
```

```
        ProxyPass /seafhttp ws://localhost:8082/seafhttp
        ProxyPassReverse /seafhttp ws://localhost:8082/seafhttp

    # --- CORS pour OnlyOffice et rclone
    <Location /seafdav>
        Header always set Access-Control-Allow-Origin "*"
        Header always set Access-Control-Allow-Methods "GET, POST, PUT, DELETE, OPTIONS"
        Header always set Access-Control-Allow-Headers "Authorization, Content-Type"
        Header always set Access-Control-Allow-Credentials "true"
    </Location>

    SSLCertificateFile /etc/letsencrypt/live/drive.miraceti.net/fullchain.pem
    SSLCertificateKeyFile /etc/letsencrypt/live/drive.miraceti.net/privkey.pem
    Include /etc/letsencrypt/options-ssl-apache.conf
    </VirtualHost>
```

office.miraceti.net-le-ssl.conf

```
<VirtualHost *:443>
        ServerName office.miraceti.net

    ErrorLog ${APACHE_LOG_DIR}/office.miraceti.net.error.log
    CustomLog ${APACHE_LOG_DIR}/office.miraceti.net.access.log combined

    # ===== En-têtes HTTPS et Proxy =====
    RequestHeader set X-Forwarded-Proto "https"
    RequestHeader set X-Forwarded-Ssl "on"

    ProxyPreserveHost On
    ProxyRequests Off
    SSLProxyEngine on

    # ===== Gestion CORS =====
    <Location />
        Header always set Access-Control-Allow-Origin "https://drive.miraceti.net"
        Header always set Access-Control-Allow-Methods "GET, POST, PUT, OPTIONS, DELETE"
        Header always set Access-Control-Allow-Headers "Authorization, Content-Type, Accept,
Origin, Referer, User-Agent"
        Header always set Access-Control-Allow-Credentials "true"
    </Location>
```

```
    # ===== Rediriger les WebSockets =====
    # (WebSocket = wss:// = port 443)
#    ProxyPassMatch "^/([0-9a-zA-Z\.\-]+)/doc/[0-9a-f\-]+/c/(.*)/websocket$"
"ws://127.0.0.1:8081/$1/doc/$2/websocket"
    ProxyPassMatch "^/(.*)/websocket$" "ws://127.0.0.1:8081/$1/websocket"

    # ===== Reverse Proxy classique =====
    ProxyPass / http://127.0.0.1:8081/
    ProxyPassReverse / http://127.0.0.1:8081/

    # ===== Sécurité SSL =====
    SSLEngine on
SSLCertificateFile /etc/letsencrypt/live/office.miraceti.net/fullchain.pem
SSLCertificateKeyFile /etc/letsencrypt/live/office.miraceti.net/privkey.pem
Include /etc/letsencrypt/options-ssl-apache.conf
</VirtualHost>
```

Revision #7
Created 9 October 2025 14:42:25 by gpatruno
Updated 11 October 2025 17:16:47 by gpatruno